OFW 2019: An Introduction to Discrete Adjoint Optimization in OpenFOAM

Markus Towara and Uwe Naumann

Informatik 12: Software and Tools for Computational Engineering, RWTH Aachen University

Instructions

(see also slide deck)

1. Install Workshop environment

Choose one of the following options:

1.1. **Install Binaries.** Binaries from debian stretch with gcc-6, may or may not work on other distributions

- cd into a workspace folder of your liking
- `wget https://stce.rwth-aachen.de/files/ofw19_binary.tar.gz`
- `tar -xzf ofw19_binary.tar.gz`
- `source discreteAdjointOpenFOAM-plus/etc/bashrc`

1.2. **Install the Docker environment.** If you do not have docker installed, please follow the steps outlined in
`https://docs.docker.com/install/linux/docker-ce/ubuntu/`
If you can't or don't want to install docker, see Section 8 for alternatives.

First we will set up the docker environment:

- cd into a workspace folder of your liking
- `wget https://stce.rwth-aachen.de/files/ofw19_docker.tar.gz`
- `tar -xzf ofw19_docker.tar.gz`
- if your user is not in the `docker` group you may need to run some of the following commands with `sudo`
- `./create` to create docker container from image (only needed once)
- folder `tutorial_data` will be mounted within the container as `$HOME`
- `./run` to run and attach to `/bin/bash` within container
- `/opt/discreteAdjointOpenFOAM-plus/etc/bashrc` will be sourced on startup by `~/.bashrc` in adjoint mode `DOF_AD_MODE=A1S`
- `pyFoam` is available within the container
- you can type `exit` to close the container again
- run `paraView` and other tools which neeed GUI (e.g. editors) on your local machine

1.3. **Install VirtualBox VM.** Binaries and tutorial data within a minimal XUbuntu 18.04 LTS. Requires Oracle VirtualBox.

- cd into a workspace folder of your liking
- `wget https://stce.rwth-aachen.de/files/ofw19_vm.tar.gz`
- `tar -xzf ofw19_vm.tar.gz`
- execute the VM with VirtualBox

2. adjointSimpleFoam

We start with the easiest possible, if slightly impractical, adjoint solver:

- `cd $OFW_DATA/adjointSimpleFoam`
- `wmake`
- `cd referenceCase`
- inspect and run `./Allrun`
- if you run out of RAM reduce the numer of time steps in `system/controlDict`
- disable `SDLS` on line 20 of `system/fvSolution`
- whats the impact to peak memory?

3. adjointSimpleCheckpointingFoam

To reduce the memory impact we employ checkpointing:

- `cd $OFW_DATA/adjointSimpleCheckpointingFoam`
- `wmake`
- `cd referenceCase` or `cd pitzDaily`
- inspect and run `./Allrun`
- try checkpointing methods `equidistant` and `revolve` in `system/checkpointingDict`
- alter the number of checkpoints
- whats the impact on run time?
- optional: try to run the solver in parallel

4. Topology optimization with piggyOptSimpleFoam

Up until now we only calculated sensitivities. Lets use them to run an optimization:

- `cd $OFW_DATA/piggyOptSimpleFoam`
- `wmake`
- `cd filter_case`
- inspect and run `./Allrun`

5. Shape optimization with piggyShapeSimpleFoam

Instead of using a penalty field, lets use all points on the boundary as parameters:

- inspect `$FOAM_SRC/OpenFOAM/meshes/polyMesh/polyMesh.C`
- `cd $OFW_DATA/piggyShapeSimpleFoam`
- `wmake`
- `cd cylinderMirror`
- inspect and run `./Allrun` and `./optimize.sh`
- enable or disable the volume constraint in `system/fvSolution`

6. Implement flow uniformity cost function

Lets try to optimize for flow uniformity. For simplicity we can assume that the mesh is uniform (orthogonal, cell volume constant).

- `cd $OFW_DATA/flowUniformity`
- implement J_U such that $J_U = (\|\bar{U}\|_0 - \|\bar{U}\|_1)^2$ where $\|\bar{U}\|_0$ and $\|\bar{U}\|_1$ are the average velocities on `outlet0` and `outlet1` respectively. Hint: Use `gAverage` and use `phi` as a substitute for `U`.
- `wmake`
- go to `cd flow_uniformity_case` and run

Questions after the Workshop? Contact: towara@stce.rwth-aachen.de